

# Image Browsing and Natural Language Paraphrases of Semantic Web Annotations

Christian Halaschek-Wiener, Jennifer Golbeck, Bijan Parsia,  
Vladimir Kolovski, and Jim Hendler

Maryland Information and Network Dynamics Lab, 8400 Baltimore Ave.  
College Park, MD, 20740 USA  
{halasche,golbeck,kolovski,hendler}@cs.umd.edu  
bparsia@isr.umd.edu

**Abstract.** Recently, there has been interest in marking up digital images with annotations describing the content of the images using Web based ontologies encoded in the W3C's Web Ontology Language, OWL. The annotations are subsequently exploited to improve the user experience of large collections of images, whether by enhance search or by a structured browsing experience. In the latter case, the complexity and unfamiliarity of logic-based ontology languages may do more to impede, than aid, the user. To alleviate this problem, we propose using automatic generation of natural language (NL) paraphrases of OWL statements to assist browsing image content. In this paper, we provide an overview of our NL generation approach and an empirical evaluation of the use of our paraphrases for image browsing.

## 1 Introduction

Recently there has been interest in using Semantic Web ontologies encoded in the Web Ontology Language [3], OWL, to formally represent the semantic content of digital images.[1]. Given semantically rich image metadata, collections can be more accurately searched and browsed, with new knowledge derived from existing annotations. Additionally, by exploiting standard Web mechanisms, one is able to link existing image collections to arbitrary knowledge repositories and vice versa.

The canonical OWL interchange syntax is based on the XML serialization of RDF (RDF/XML). Neither RDF nor XML were designed readability in mind (much less casual end user readability). There are several alternative surface syntaxes designed with more readability in mind [14, 15] and, due to the correspondence with first order logic, there are a number of traditional logic notations available. However, these all require a fairly deep understanding of OWL, logic, or both which, aside from being comparatively rare, seems unnecessary for the purpose of navigating through image collections. Recently, there has been work in providing natural language translations of OWL concept definitions, providing users with a format that is easier to both read

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE <b>2006</b>		2. REPORT TYPE		3. DATES COVERED <b>00-00-2006 to 00-00-2006</b>	
4. TITLE AND SUBTITLE <b>Image Browsing and Natural Language Paraphrases of Semantic Web Annotations</b>				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>University of Maryland, Maryland Information and Network Dynamics Laboratory, College Park, MD, 20742</b>				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>					
13. SUPPLEMENTARY NOTES <b>The original document contains color images.</b>					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES <b>12</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			

and understand [2, 6, 8, 9]. In [2], we provided an algorithm for paraphrasing OWL concepts with a controlled natural language.

In this paper, we propose the use of automatically generated NL paraphrase of OWL classes and individuals to aid in browsing image. We feel that the paraphrases will make allow users to more effectively use and enjoying using the semantic annotations when browsing image collections, and to better understand the content of the browsed images.

Most semantic markup of images includes a large number of instance assertions, corresponding to features of concrete particular depicted in the images. Therefore, we extend the algorithm in [2] to provide natural language paraphrases of OWL assertions about individuals, as well as of class definitions. We have implemented this algorithm as a plug-in to an ontology-based, image annotation and browsing tool, PhotoStuff [1]. Lastly, we provide an empirical evaluation of the effectiveness of these NL paraphrases for the task of browsing and interacting with image collections.

## 2 Background

[2] presents an approach for automatically generating natural language paraphrases for OWL concept definitions that attempts to preserve the underlying meaning of the logic based definition. The approach is applicable to any ontology where the classes and properties are named using certain standard naming conventions. The algorithm is fairly straightforward considering the good quality results achieve. The most sophisticated NL processing tool the algorithm uses is a part-of-speech (POS) tagger, which is used to improve the fluency of the generated NL description.

The first step in the approach is to generate a parse tree corresponding to the relations between the OWL class and other entities. Creating the tree provides additional flexibility to alter it (post-process) in any way deemed necessary (a sample parse tree is provided in [2]).

In [2], it has been noted that property names from several major ontologies reveal that, while properties could theoretically be named with arbitrary words, their names are generally parsed into one of a small number of simple phrase structures. These structures can algorithmically be restructured, by using a POS tagger, providing a (more) natural language style format. Below, Table 1 lists these phrase structure categories, along with their reformatted natural language translations.

**Table 1.** Common class and property phrase structure, along with natural language translations

1. (has) NP
  - Examples: email, hasColor
  - Expansions: X has a color Y
  - Alternate (if Y is an AdjP): X has Y color
2. V
  - Example: knows
  - Expansion: X knows Y
3. (is) NP P
  - Examples: brotherOf, isBrotherOf
  - Expansion: X is a brother of Y
4. (is) VP P
  - Examples: producedBy, isMadeFrom
  - Expansions: X is produced by Y, X is made from Y
5. VP NP
  - Example: producesWine
  - Expansion: X produces a wine Y
  - Alternate (if Y is an AdjP): X produces a Y wine
6. is NP
  - Example: isMetal
  - Expansion: X is a metal
  - Alternate (boolean value is false): X is not a metal
7. (is) AdjP
  - Example: isHardWorking
  - Expansion: X is hard working
  - Alternate (boolean value is false): X is not hard working

After generating the parse tree, there are several steps in generating the NL output, including a pre-processing step where the tree is modified to eliminate nodes containing *owl:Thing*, etc. Further details are available in [2]. In general, the approach generates full English sentences whenever possible. However, we have found that rendering complex concepts entirely in NL sometimes results in very lengthy, difficult to understand sentences. In some cases, using a bulleted, nested list format for such complex sets of conditions was much clearer. For example, part of the definition of *Beaujolais* from the wine ontology<sup>1</sup> is given below in Table 2.

**Table 2.** Bulleted natural language rendering of OWL class *Beaujolais*

- A Beaujolais is a Wine that:
- is made from at most 1 grape, which is Gamay Grape
  - has Delicate flavor
  - has Dry sugar
  - has Red color
  - has Light body

---

<sup>1</sup> Wine OWL Ontology: <http://www.w3.org/2001/sw/WebOnt/guide-src/wine.rdf>

### 3 OWL Individual NL Paraphrases

We can categorize images based on the types of things they depict. In the SemSpace<sup>2</sup> portal, a nested list rendering of the class hierarchy is the initial interface to the collection. Thus, NL paraphrases of concepts might help users determine which categories are likely to contain images of interests. However, images, especially photographs, generally depict concrete objects that are naturally represented in OWL as individuals with various types and properties. Thus, if paraphrases are to help the user understand the contents of particular images, we must extend the algorithm in [2] to handle NL paraphrases of OWL individuals as well.

#### 3.1 Approach Overview

The approach adopted here provides an NL rendering of OWL individuals based on the direct relations of that individual (e.g., type assertions, labels, defined relations, etc.). In order to generate an NL paraphrase for an OWL individual, first a NL parse tree is generated for that individual [2]. In contrast to the approach for OWL concepts [2], this tree will be at most one level deep, as only the direct relations are used. As the tree is created, only *rdf:type* assertions corresponding to the (possibly inferred) most specific classes of the individual are added as edges. (In certain ontologies with shallow, informative class graphs, it might be preferable to add all the types of the individuals, or to control the depth in a different way.) The relation edges are labeled by two distinct mechanisms: An *rdf:type* edge in the tree is given an “is a” label. For all additional relations, the POS tagger is run over the “local part” of the property’s URI (as in [2]) to provide more legible labels for the relations. Finally, the labels for the objects of the relations are added to the object nodes. Figure 1 depicts a subset of the NL parse tree and the original RDF/XML for a sample individual, *Storey Musgrave*.

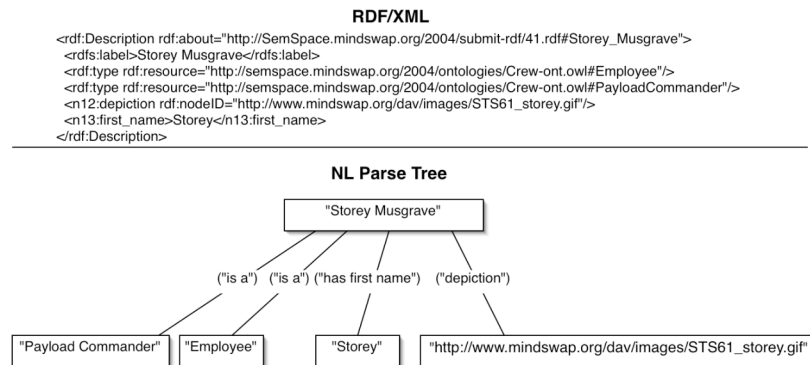


Fig. 1. Subset of RDF/XML and of NL Parse Tree for OWL Individual Storey Musgrave

<sup>2</sup> <http://semSPACE.mindswap.org/>

After this post-processing the NL rendering is generated. The current approach uses the following template for producing OWL individual NL renderings: First the label edges of the NL tree for the given individual are retrieved. This label is used to begin the NL sentence. Following this, a comma separated rendering of all *rdf:type* edges is generated. Then a bullet list of all the additional relations is produced. The template is shown below in Table 3.

**Table 3.** RDF/XML serialization of individual Storey Musgrave vs. the automatically generated natural language rendering. Italicized words are invariant of the individual being rendered

```
Individual is a rdf:type that
- relation 1
- relation 2
- ...
- relation n
```

For example, the NL rendering for the OWL individual *Storey Musgrave* is presented below in Table 4 (note that the original RDF/XML for Storey Musgrave is provided in Table 3).

**Table 4.** Automatically generated natural language rendering of Storey Musgrave

#### NL Rendering

```
Storey Musgrave is a Payload Commander, Employee that
- has first name "Storey"
- has last name "Musgrave"
- depiction musgrave_sts61, STS033, musgrave , sts-61crew,
  STS061_storeyL, STS61_storey, musgrave WFPC
- has space flights "sts-61", "sts-33"
- has date of birth "08/19/1935"
```

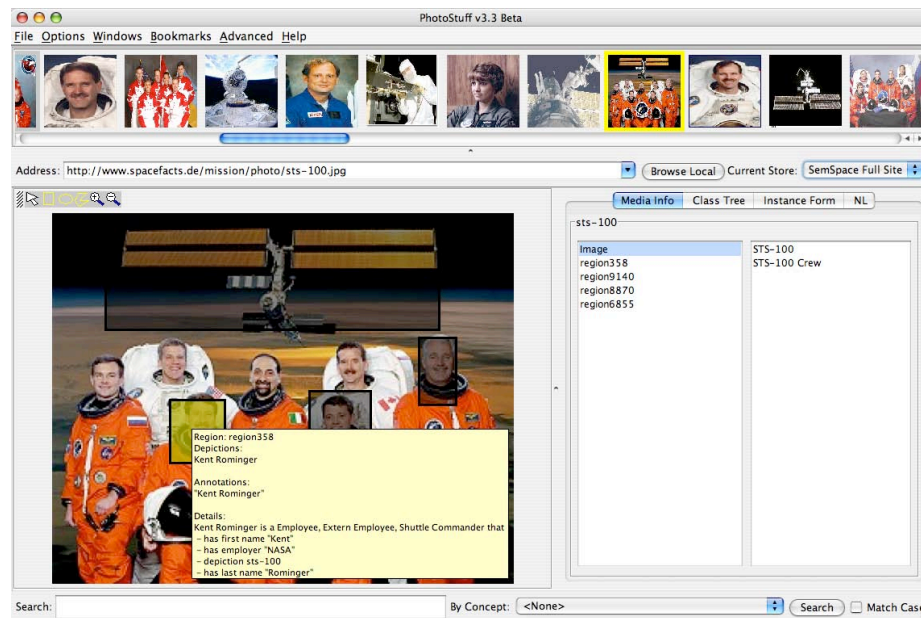
## 4 NL Captions for Browsing Image Collections

In this section we provide details of using automatically generated NL paraphrases for browsing image collections in our image annotation tool, PhotoStuff [1]. The main motivation for using NL paraphrases for browsing image collections and their annotations is that it will provide a more enjoyable user experience. Essentially, the user will be provided with the details about image contents in a human readable and understandable format, which allows them to focus on the images and what they depict rather than the cryptic details of the representation language. One additional benefit of publishing the natural language rendering of the images is that it can allow exiting search engine technology to index such content.

## 4.1 Implementation Details

PhotoStuff is a platform independent, open source, image annotation tool that allows users to annotate images and their sub-regions using concepts from any number of ontologies specified in OWL. PhotoStuff can also load pre-existing annotations, which can then be browsed or used in subsequent annotations.

An experimental implementation of our NL generation algorithm has been provided as a plug-in for PhotoStuff. Within PhotoStuff, the NL captions are used in two main ways. First, when an image is selected and loaded into the image canvas, NL renderings for the instances depicted within regions are provided as pop-ups when the regions are moused over (see Figure 2). This provides the user with a quick, human readable display of the individual depicted within that region.



**Fig. 2.** Region NL Caption Pop-Up

Additionally, the NL for individuals depicted within regions can be viewed by right-clicking the regions, and selecting "View NL". This puts the natural language paraphrases in a NL info pane, as shown below in Figure 3. The figure illustrates that in the NL info pane, the generated natural language includes hyperlinks for all additional individuals and OWL class that occur in the rendering of the current individual or class. This allows the user to browse the image collection based on the existing annotations in a Web-like manner. Further, the hyperlinks can be right clicked, providing the option to filter the thumbnail strip to only show images that depict that instance or instances of that class.

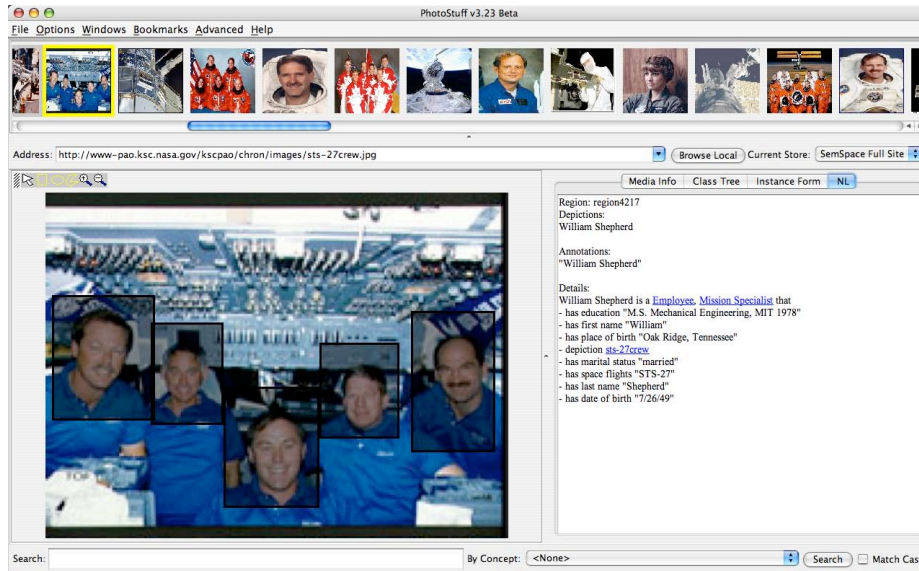


Fig. 3. NL Info Pane for Browsing Image Annotations in NL Format

## 5 Evaluation

To evaluate the benefits of the NL format, we conducted a pilot user study. Our hypothesis was that users would prefer the NL format for viewing data when the task was to gain an understanding of the meaning of a concept.

Our pilot study included seven subjects. Subjects ranged in age from 20 to 37 and all were students working toward bachelors, masters, or Ph.D. degrees. We tested the subjects' preferences for method of viewing classes separately from viewing instances. For both classes and instances, we choose three examples: one very simple, one of medium complexity, and one very complicated example.

When viewing classes, we used the Wine ontology mentioned earlier. Our simple class is *AlsatianWine*, which only had one restriction. *Anjou* was the medium complexity class, with four simple restrictions and an intersection combined with a restriction. The most complex class was *Beaujolais*, with six simple restrictions, including cardinality, and an intersection combined with a restriction.

Each class was presented to the subjects in SWOOP [13]. For the study, four additional formats for viewing each class was provided in SWOOP (see Figure 4).



<p>[[[ Source: Main SWOOP ]]]</p> <p><b>Time:</b> 2006-02-07 23:22:03  <b>Ontology:</b> koala.owl</p> <p><b>OWL-Class:</b> <a href="#">Koala</a></p> <p><b>Subclass of:</b>  <a href="#">(3hasHabitat , DryEucalyptForest)</a>  <a href="#">(3isHardWorking , { "false"^^&lt;xsd:boolean&gt; })</a>  <a href="#">Marsupials</a></p> <p>Namespace(dc = &lt;http://purl.org/dc/elements/1.1/&gt;)  Namespace(xsd = &lt;http://www.w3.org/2001/XMLSchema#&gt;)  Namespace(rdf = &lt;http://www.w3.org/1999/02/22-rdf-syntax-ns#&gt;)  Namespace(owl = &lt;http://www.w3.org/2002/07/owl#&gt;)  Namespace(koala = &lt;http://protege.stanford.edu/plugins/owl/owl-library/koala.owl#&gt;)  Namespace(rdfs = &lt;http://www.w3.org/2000/01/rdf-schema#&gt;)</p> <p>Ontology( &lt;http://protege.stanford.edu/plugins/owl/owl-library/koala.owl#&gt;  Class (koala:Koala partial  restriction(koala:hasHabitat someValuesFrom( koala:DryEucalyptForest))  restriction(koala:isHardWorking value ( "false"^^&lt;http://www.w3.org/2001/XMLSchema#boolean&gt; ))  koala:Marsupials  )</p>	<p>[[[ Source: Main SWOOP ]]]</p> <p><b>Time:</b> 2006-02-07 23:23:34  <b>Ontology:</b> koala.owl</p> <p>@prefix xsd: &lt;http://www.w3.org/2001/XMLSchema#&gt; .  @prefix owl: &lt;http://www.w3.org/2002/07/owl#&gt; .  @prefix rdfs: &lt;http://www.w3.org/2000/01/rdf-schema#&gt; .  @prefix : &lt;http://protege.stanford.edu/plugins/owl/owl-library/koala.owl#&gt; .</p> <pre> :Koala a owl:Class;   rdfs:subClassOf     [ a owl:Restriction;       owl:onProperty :hasHabitat;       owl:someValuesFrom :DryEucalyptForest     ];   rdfs:subClassOf     [ a owl:Restriction;       owl:onProperty :isHardWorking;       owl:hasValue "false"^^xsd:boolean     ];   rdfs:subClassOf :Marsupials . </pre> <pre> &lt;rdf:RDF xml:base="koala#"   xmlns:owl="owl;"   xmlns:rdf="rdf;"   xmlns:rdfs="rdfs;"&gt;   &lt;owl:Class rdf:about="#koala"&gt;     &lt;rdfs:subClassOf&gt;       &lt;owl:Class rdf:about="#Marsupials"/&gt;     &lt;/rdfs:subClassOf&gt;     &lt;rdfs:subClassOf&gt;       &lt;owl:Restriction&gt;         &lt;owl:onProperty&gt;           &lt;owl:ObjectProperty rdf:about="#hasHabitat"/&gt;         &lt;/owl:onProperty&gt;         &lt;owl:someValuesFrom&gt;           &lt;owl:Class rdf:about="#DryEucalyptForest"/&gt;         &lt;/owl:someValuesFrom&gt;       &lt;/owl:Restriction&gt;     &lt;/rdfs:subClassOf&gt;     &lt;rdfs:subClassOf&gt;       &lt;owl:Restriction&gt;         &lt;owl:hasValue rdf:datatype="xsd:boolean"&gt;false&lt;/owl:hasValue&gt;         &lt;owl:onProperty&gt;           &lt;owl:DatatypeProperty rdf:about="#isHardWorking"/&gt;         &lt;/owl:onProperty&gt;       &lt;/owl:Restriction&gt;     &lt;/rdfs:subClassOf&gt;   &lt;/owl:Class&gt; &lt;/rdf:RDF&gt; </pre>
---	--

**Figure 4.** Different view renderings used for evaluation of class descriptions. Clock-wise from upper left: Concise format, Turtle, Abstract Syntax, and RDF/XML.

Users were asked to view the concept with the goal of understanding it's meaning, rather than the modeling features. They were allowed to take as much time as they needed and, when finished, they were asked to rank the formats according to their preference, with 1 being best and 5 being worst. If the subjects felt two formats were equally good, they were allowed to give them the same ranking.

For all three classes, the order of the rankings was the same. From best to worst, the formats were ranked, and shown below in Table 5.

**Table 5.** Rankings from best to worst of view format of class descriptions

1. Natural Language
2. Concise Format
3. Abstract Syntax
4. Turtle
5. RDF/XML

Additionally, the average ranking and standard deviations for the various formats is presented below in Table 6.

**Table 6.** Average rankings and standard deviations for the five formats used to display class information

Class	Average Rank (Standard Deviation)				
	Concise Format	Abstract Syntax	Natural Language	RDF/XML	Turtle
<b>AlsatianWine</b>	2.00(1.41)	3.57(1.13)	1.57(0.79)	4.86(0.38)	3.86(0.69)
<b>Anjou</b>	1.86(0.69)	2.71(0.49)	1.14(0.38)	5.00(0.00)	4.00(0.00)
<b>Beaujolais</b>	2.14(0.69)	2.71(0.49)	1.00(0.00)	5.00(0.00)	3.86(0.38)

Using a Wilcoxon matched-pairs signed-ranks test, we found that the NL format significantly outperformed the Concise Format (ranked second on average) for both Anjou and Beaujolais with  $p < 0.05$ . There was not a significant benefit over the Concise format for the simplest class, AlsatianWine, but NL did significantly outperform the Abstract Syntax, which was the average third ranked format. This allows us to conclude that in the pilot study the NL format offers significant benefits to users when they are trying to understand the meaning of classes, particularly complex classes.

To evaluate the NL format with instances, subjects were given three instances from the collection of astronaut data<sup>3</sup>. The first instance, Bryan O'Conner, had only one property: the depiction that tied him to an image. The second instance, John Young, had a depiction and one additional property. The last instance, Storey Musgrave, was tied to three regions and four properties, including both Datatype and Object properties. Subjects viewed the data about each instance in a tabular format and in the NL format with the goal of understanding the information about each instance. Subjects took the time they needed without limits and then were asked to choose which format they preferred.

For each of the three instances, the NL format was preferred 6:1 over the instance form. When subjects commented on their preference, it focused largely on the fact that the NL format was more concise, displaying only relevant information.

We conclude this section by stating that the results of this pilot study show that users feel that the NL format provides them with an advantage when trying to understand both classes and instances.

## 6 Discussion and Future Directions

While our initial evaluations support the usefulness of the approach presented here, we note there are a few limitations, which we leave as future work. First, if individuals participate in a large number of relations, then the generated bullet list, or con-

<sup>3</sup> SemSpace portal instance data. Available at <http://semspace.mindswap.org/rdf/dump>

catenated sentence, can be quite long and cumbersome to read through. One potential solution to ease this problem is to use a ranking metric when displaying the relations in the NL generated format. [4, 5] propose relationship-ranking metrics that could potentially be used to filter irrelevant relations, thus presenting the user with a smaller NL paraphrase.

An additional solution to this problem we would like to investigate is the to utilize text summarization techniques for the output from the NL generation algorithm. [10, 11, 12] present a variety of techniques for summarizing such textual data. Solutions such as these could possibly be applied to the NL output, thus presenting the user with a succinct version of the output from the NL generation engine.

In this work, we use a predefined template view of the NL rendering of instances. Under some circumstances, particular users may wish to define their own templates. Thus we would like to support using custom templates for particular ontologies or class descriptions.

In our pilot study, we focused on user understanding of OWL data and found, generally, that users preferred the NL paraphrases for that purpose. While we found this format to be preferred for understanding image annotations, the extent of which these paraphrases work to improve the overall image browsing experience remains to be seen. Therefore we plan to perform further evaluations of the approach.

## 7 Related Work

In this section we present related work in the area of automatically generating natural language translations of Semantic Web ontological concepts and instances. [6] provides an instructional use of NL paraphrases for understanding OWL Concepts. The authors mention a plug-in to Protégé, the *Class Description Display*<sup>4</sup> plug-in that provides simpler NL descriptions that resemble OWL Abstract Syntax. This work has been refined into the so-called Manchester syntax.<sup>5</sup> In [7], a subset of English is introduced called Attempto Controlled English (ACE). ACE is translated into first-order logic and thus can be used as a formal notation. Therefore, ACE is a formal language with the semantics of first order logic. In comparison, our approach converts OWL classes and individuals, which are based on a decidable subset of FOL called Description Logics, into a NL description. [8] describes an XML-based NL generation for RDF and DAML+OIL. In this work, a pipeline of XSLT transformations implements the sequence of processing stages in the orthodox pipeline architecture for NL generation. The generator uses predefined XSLT text plan templates for specific ontologies, following a domain-specific approach of shallow generation. However, it is unclear whether this approach works efficiently for more complex OWL ontologies. [9] presents an approach for automatic generation of reports from OWL ontologies, using natural language generation tools. Our work here differs in that we do not rely on a lexicon for NL generation.

---

<sup>4</sup> Class Description Display plug-in. Available at <http://www.co-ode.org/downloads/cdc/>

<sup>5</sup> [http://www.co-ode.org/resources/reference/manchester\\_syntax/](http://www.co-ode.org/resources/reference/manchester_syntax/)

## 7 Conclusions

In this paper we have provided an extension of our previous work [2] to automatically generate natural language paraphrases for OWL individuals. Given this, we have proposed using this approach for browsing Semantic Web annotations of image collections. This provides the user with a more pleasant experience of the annotations.

We have additionally presented an experimental implementation of the approach in an ontology based, image annotation tool, PhotoStuff. Finally, we have provided an empirical evaluation of the usage of the NL paraphrases for browsing image collections, finding that it was in fact useful for the task.

This work was supported in part by grants from Fujitsu, Lockheed Martin, NTT Corp., Kevric Corp., SAIC, the National Science Foundation, the National Geospatial-Intelligence Agency, DARPA, US Army Research Laboratory, and NIST. We would also like to thank Aditya Kalyanpur and Mike Grove for all of their contributions to this work.

## References

1. Halaschek-Wiener, C., Schain, A., Golbeck, J., Grove, M., Parsia, B., Hendler, J.: A Flexible Approach for Managing Digital Images on the Semantic Web, 5th International Workshop on Knowledge Markup and Semantic Annotation, Galway, Ireland, November 7, 2005.
2. Hewlett, D., Kalyanpur, A., Kolovski, V., Halaschek-Wiener, C.: Effective Natural Language Paraphrasing of Ontologies on the Semantic Web. End User Semantic Web Interaction Workshop, International Semantic Web Conference (ISWC), November 2005, Galway, Ireland.
3. Dean, M., Schreiber, G.: OWL Web Ontology Language Reference. W3C Recommendation. February 2004. <http://www.w3.org/TR/2004/REC-owl-semantics-20040210/>
4. Aleman-Meza, B., Halaschek-Wiener, ., Arpinar, B., Ramakrishnan, C., Sheth, A.: A Flexible Approach for Analyzing and Ranking Complex Relationships on the Semantic Web, IEEE Internet Computing (Journal), Special Issue on Information Discovery: Needles and Haystacks, May-June 2005.
5. Anyanwu, K., Maduko, A., Sheth, A.: SemRank: Ranking Complex Relationship Search Results on the Semantic Web. The 14th International World Wide Web Conference, (WWW2005), Chiba, Japan, May 10-14, 2005
6. Rector, A., Drummond, N., Horridge, M., Rogers, J., Knublauch, H., Stevens, R., Wang, H., Wroe, C.: OWL Pizzas: Common errors & common patterns from practical experience of teaching OWL, European Knowledge Acquisition Workshop (EKAW) 2004.
7. Fuchs, N., Schwertel, U., Torge, S.: Controlled natural language can replace first-order logic. 14th IEEE International Conference on Automated Software Engineering, Cocoa Beach, Florida.

8. Wilcock, G.: Talking owls: Towards an ontology verbalizer. Proceedings of the 2nd International Semantic Web Conference (ISWC), Oct. 2003.
9. Bontcheva, K., Wilks, Y.: Automatic Report Generation from Ontologies: the MIAKT approach, In Proceedings Ninth International Conference on Applications of Natural Language to Information Systems (NLDB'2004), Manchester, UK.
10. Buyukkokten, O, Garcia-Molina, H., Paepcke, A.: "Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices". The 10th International WWW Conference (WWW10). Hong Kong, China - May 1-5, 2001.
11. Barzilay, R., Elhadad, M.: Using Lexical Chains for Text Summarization. In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization, pages 10–17, Madrid, Spain, August 1997.
12. Hovy, E.H. Lin, C.: Automated Text Summarization in SUMMARIST. In M. Maybury and I. Mani (eds), Advances in Automatic Text Summarization. Cambridge: MIT Press. 1999.
13. Aditya Kalyanpur, Bijan Parsia, Evren Sirin, Bernardo Cuenca-Grau, and James Hendler. Swoop - a web ontology editing browser. Journal of Web Semantics, 4(1), 2005.
14. Beckett, D., Turtle — Terse RDF Triple Language, <<http://www.dajobe.org/2004/01/turtle/>>, February, 2006.
15. Patel-Schneider, P., Hayes, P., Horrocks, I., OWL Web Ontology Language Semantics and Abstract Syntax, <<http://www.w3.org/TR/owl-semantics/>>, February, 2004.